Open Problems of Trustworthiness and Trust in Autonomous Systems

Natalia M. Alexandrov

NASA Langley Research Center

IMA, January 2023

Content

- Motivating settings
- A sample of difficult and timely problems in the area of trustworthiness/trust of autonomous systems that would benefit from mathematical attention in modeling and optimization
- Trustworthiness: “Mind”
- Trustworthiness: “Body”

Definition of "Autonomous System"

- Extensive arguments on definitions and meaning of autonomy

- <u>My</u> definition:
  - Autonomous System $\equiv$ Cyber(-physical-human) system with a capacity for <u>independent</u> decision making and <u>authority</u> to act on decisions in a <u>specified environment</u>

- E.g., my vacuum cleaner is an autonomous system

Overarching Question

- When can we (justifiably) trust an autonomous system in safety-critical and time-critical environments, to inform certification and public trust?

CPH Trust/Trustworthiness Breakdowns

Problem Setting: safety-critical, time-critical
Dependent on certification

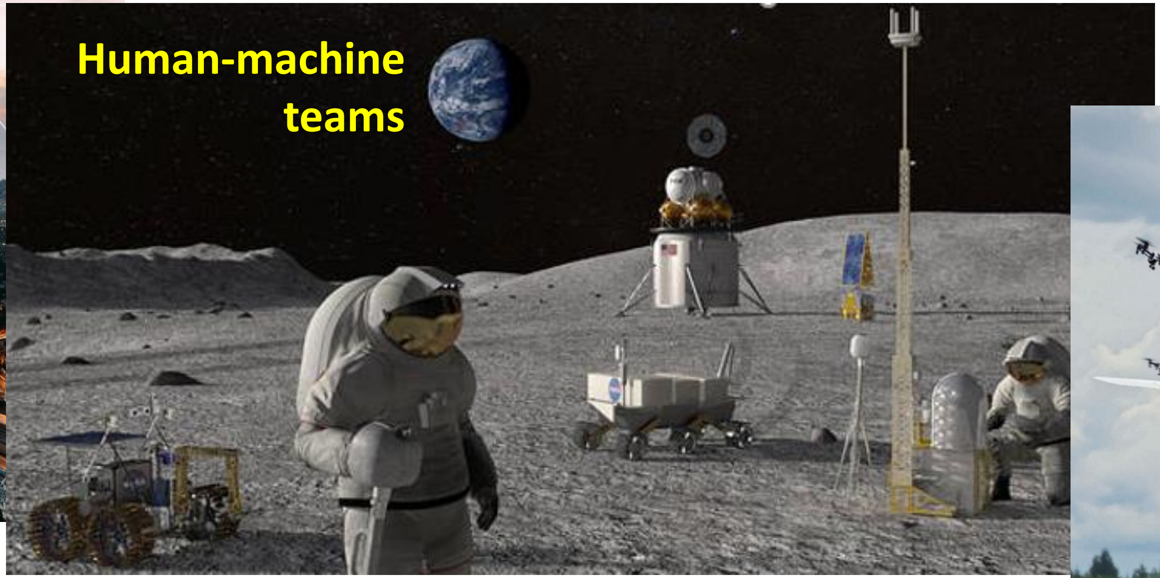


Image credits: NASA

Density + heterogeneity + autonomy + non-cooperative agents ⇒
System complexity increases / uncertainty grows / safety decreases ⇒
Control must transition to human ∪ machine, with increasing machine authority
Machine authority is a major source of uncertainty

Trustworthiness vs. Trust

Some components informing trustworthiness:
- Context
- Trustworthiness models
- Physics (e.g., trajectory planning)
- Multi-objective decision-making under uncertainty
- Anomaly detection
- V&V (stress testing)
- Persistent modsim
- XAI (for performance)
- Metrics (thresholds)

Trustworthiness
- Attribute of CPH system
- Assurance that CPH does what is required
- Necessary for safety-critical environments

Context, XAI, Metrics …

Trust
- Attribute of participants, users
- Readiness to rely on another entity

Some components informing trust:
- Context
- Trustworthiness models
- Natural HMI
- Two-way learning
- Interaction history
- XAI (for transparency)
- Metrics (thresholds)

- Certification is a set of functional requirements and bounds that imply trustworthiness.
- To the best of our knowledge, there are currently no certification criteria for autonomous CPH systems.

“Mind” problems

ATTRACTOR as an example

Three-year project under NASA’s Convergent Aeronautics Solutions Project

Large multi-center team

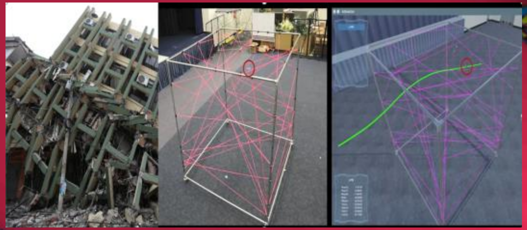
The Problem

Build a <u>basis for certification</u> of autonomous systems via establishing metrics for trustworthiness and trust in multi-agent team interactions, using AI explainability and persistent modeling and simulation, in the context of mission planning and execution, with analyzable trajectories.

Components of Complex Multi-agent Systems

Measurable **<u>trustworthiness</u>** of fundamental functional components

Context: Trajectories

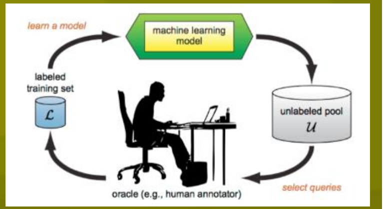


Heterogeneous multi-agent (including human-machine) teaming is essential

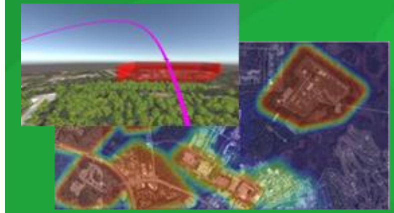
Teaming

Efficient two-way learning/ training loop.

Learning & Justifiable Trust

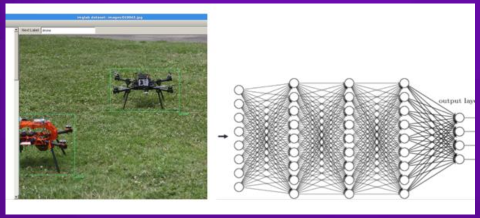


Good Outcomes: Basis for Trust

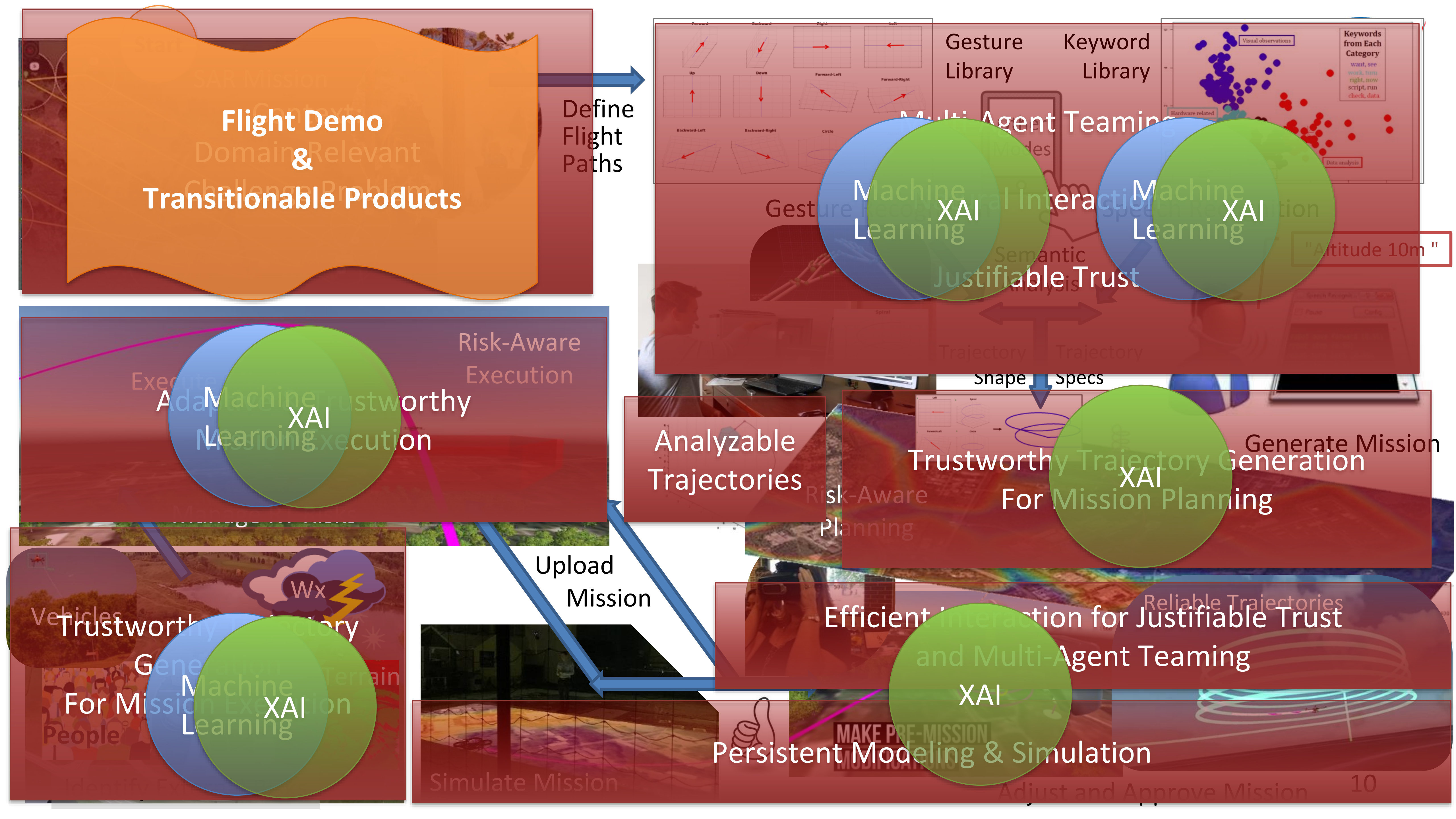


Sound multiobjective decision-making under uncertainty

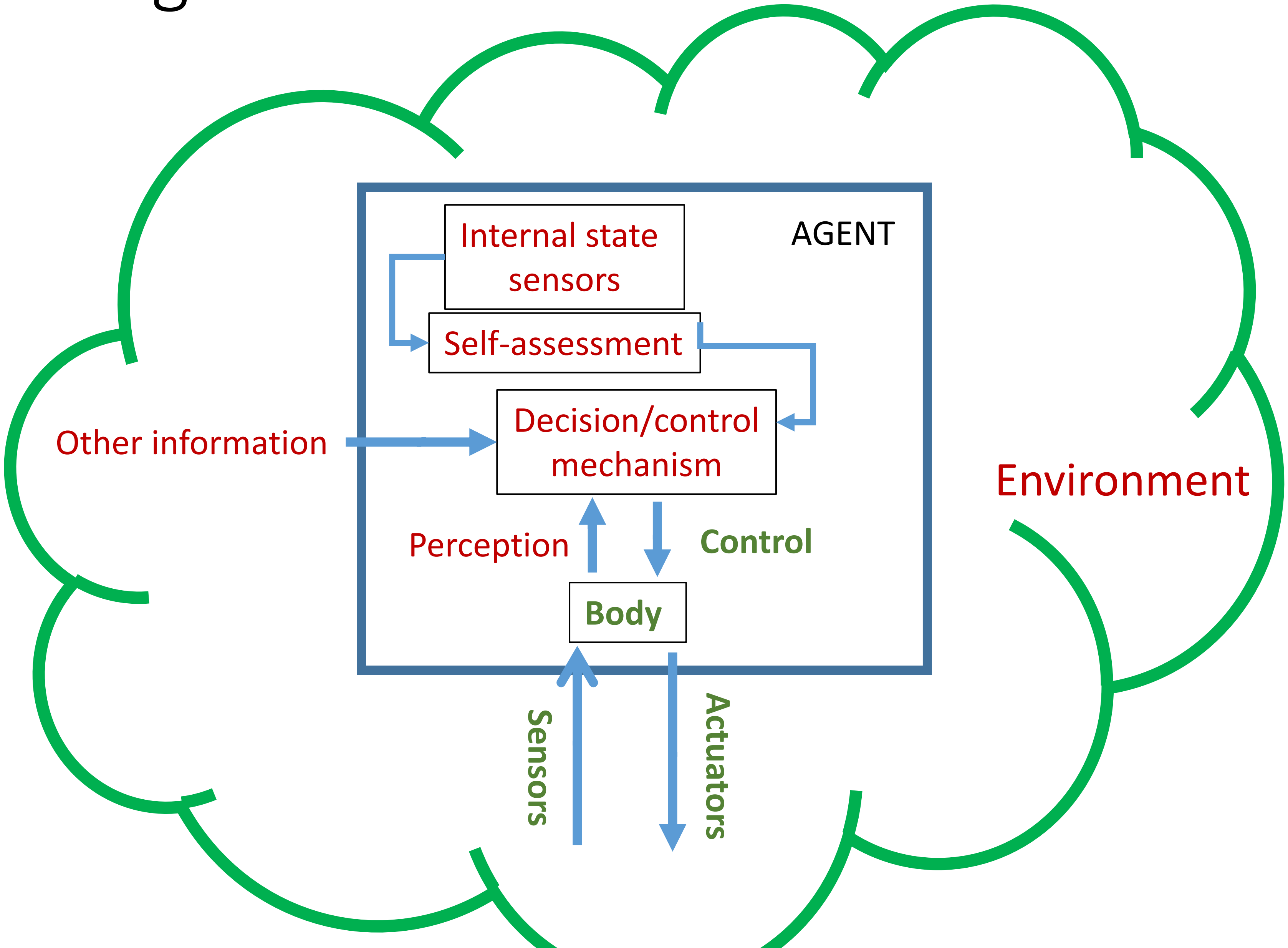
Explainability



These are two UAVs. Because… They are quadrotors. They have four propellers. They have four legs.


Flight Demo
&
Transitionable Products
Define
Flight
Paths
Gesture
Library
Keyword
Library
Multi-Agent Teaming
Machine
Learning
XAI
Justifiable Trust
"Altitude 10m"
Risk-Aware
Execution
Trustworthy
Mission Execution
Analyzable
Trajectories
Shape
Specs
Generate Mission
Trustworthy Trajectory Generation
For Mission Planning
Upload
Mission
Wx
Vehicles
Trustworthy Trajectory
Generation
For Mission Execution
People
Efficient Interaction for Justifiable Trust
and Multi-Agent Teaming
Reliable Trajectories
Persistent Modeling & Simulation
Simulate Mission
Adjust and Approve Mission

Agents



- trust $\wedge$ trustworthiness $\sim$ {reliability, efficiency, robustness, resilience, predictability of behavior, explicability, timely prediction of phase transitions, survivability} $\sim$ 1/Risk
- Conjecture:

  Green: Amenable to traditional safety-based design

  Red: New risks; new approaches needed or bound complexity and achieve trust and trustworthiness

* Red, if sensor is combined with perception.

Trust Components

- Trust = subjective probability on the part of agent A that agent B will give direction or perform an action that will result in a positive outcome (or will not result in a negative outcome) for A
- Trust = subjective expectation by A of behavior by B, based on the history of their interactions
- Trust implies dependence, reliability (trustworthiness), confidence, subjectivity, control (e.g., Kofta 2007), risk, expected benefit to the agent that exhibits trust
- In distributed systems trust is related to reputation (e.g., online systems)
- Cognitive aspects of trust (e.g., Castefranchi and Falcone 2000) are beliefs in competence, intent, persistence, dependence, realization

Evidence

- General:
  - Good decision outcomes for a long period of time
  - Explicability during training and forensics
    - Requires a shared mental model
  - Adaptability or, at least, graceful degradation in the face of unanticipated conditions
  - Recognition and warning of “no solution”
  - Prediction of phase transition from controllability to non-controllability
  - Risk minimization
- Can be accumulated via
  - Forensics, analysis, explanation
  - V&V
  - Statistically in practice (e.g., the current air traffic system)
  - Games
  - Simulations

A Sample of Working Hypotheses

- Explicability or interpretability during training, forensics and operations—except in time-critical operations—increases justified trust and trustworthiness
- Shared mental models support explicability and interpretability
- Shared mental models support sound decision making

ATTRACTOR Example: Decision Conflict

A Possible Conversation

- M: I must change a planned portion of trajectory
- H: Why?
- M: I detect children in the area. Risk rises from X to Y.
- H: Are you sure?
- M: Yes, here is the image of children.
- H: What is your new trajectory?
- M: Here is the image and associated risk.
- H: Are there alternative trajectories?
- M: Yes, but their associated risks are higher and the associated rewards are small.
- …
  - Explanations implied that the goals and risk assessment are shared
  - Q: Who has the final decision authority?
  - N.B. Representation of risk and uncertainty information to the human is a big problem (e.g., Monty Hall problem)

Shared Mental Models

Collaboration with Tufts University, Matthias Scheutz

- Mental model: a mechanism for *describing, explaining, and predicting* the behavior of the *system*
- Shared mental model: Knowledge structures shared by members of the team to enable coordination for a task and adaptation of the task (*describe, explain and predict the behavior of the team*)
- Shared models are necessary for explanation
- Tufts team:
  - Formalization of mental models (UML representation and logical notation)
  - Formalization of model similarity and compatibility
  - Extension of cognitive concepts of human teams to include software agents
  - Consider combining physical and mental models to form an “extended mind” (Rouse and Morris 1986)

Shared Mental Models

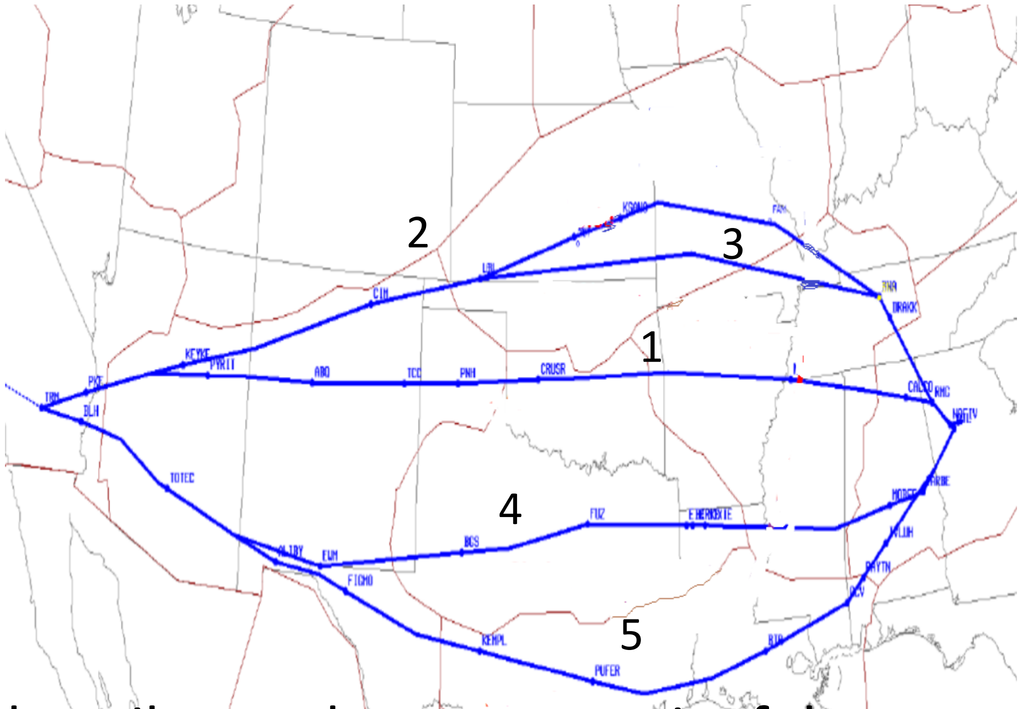
Complementary Approach in ATTRACTOR

- Claim: All decisions at all scales are solutions of optimization problems, subject to constraints, some unstated (e.g., incomplete formulation, algorithmic limitations)
- All difficulties in “Concrete Problems in AI Safety” (Amodei et al. 2017) can be traced to reward (objective) function definition
- N.B. “On the Folly of Rewarding A, While Hoping for B” S. Kerr, 1975
- View mental model as a formal optimization problem formulation
- Model similarity = shared variables, objectives, constraints, and multiobjective formalization

Proposed Approach to Explicability of Decision Making

- Consider explanations/justification/interpretation of a set of proposed trajectories:
  - Planned trajectory 1 is infeasible because violent weather is in the path of the old trajectory
  - Trajectories 2-5 are flyable and conflict-free
  - Trajectory 3 saves more fuel but flies through denser traffic that trajectory 2
  - …

Image courtesy FAA. Used for notional depiction of trajectories.



- Common features of all acceptable explanations:
  - The old trajectory violated constraints
  - New trajectories are flyable and conflict-free (satisfy constraints); objectives have better values
- Regardless of the algorithm, the trajectory has physical attributes that are meaningful to a human decision maker in terms of value functions associated with moving from point A to point B.
  - E.g., minimizing fuel expenditure, minimizing delays in reaching destination, maintaining safe distance from other aircraft, objects, and weather, etc.

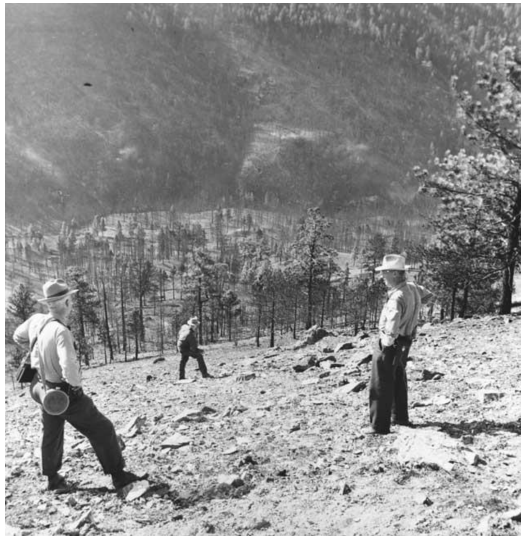
Context and Goal Driven Explanation

- Convincing explanation must contain information about
  - Constraint violations and comparative values of the objectives and constraints between the current and proposed decision, such as a trajectory
  - Sensitivity (robustness information)
  - Overall risk information (uncertainty horizons, consequences implied or explicit)
  - Information about alternative decisions via what-if scenarios if sims are available
- Note: In M-M interactions, resolving information constraints also requires physical “explanation”, appropriately represented.

Correct / Complete Objectives Support Trustworthiness

- Good adaptation
  - Robustness to distributional shift (good behavior in environments that differ from training ones)
- Bad Adaptation
  - E.g., reward hacking (if, during cleaning, you get a reward for not seeing a mess, don't look)
- Peppered moth
  - Slow
  - Not assured
  - <u>Salient point: potential variable existed, in principle</u>
- Decision making: Mann Gulch Fire
  - Want this
  - Fast
  - Not assured
  - Seemingly out of nowhere
  - <u>Salient point: potential variables/functions existed</u>

Fermi Lab



Nat. Interagency Fire Center

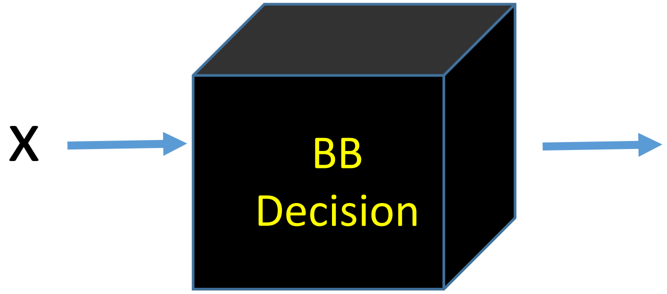
Finding Objectives

- Conjectures
  - Objectives will not arise unless they have been defined *a priori*
  - Objectives can only be "activated" or "de-activated" via adaptive adjustment of preferences during execution
  - Preference adaptation (adjustment) can be learned, given "complete" objective formulation

- In progress:
  - Establishing common "cognitive" model between human and machine decision makers for execution and explanation
  - Multiobjective preference identification for moving from point A to point B via interactive learning

Common Cognitive Model for Explanation

- Working Hypothesis: Recipients of explanations would find others' explanations convincing if they fit within their own explicability framework (although the framework may require expansion via learning)
- Local (reward) and global (utility) objectives must be interpretable to H in M-H interactions
- Initial approach to forming a common model, for both objectives



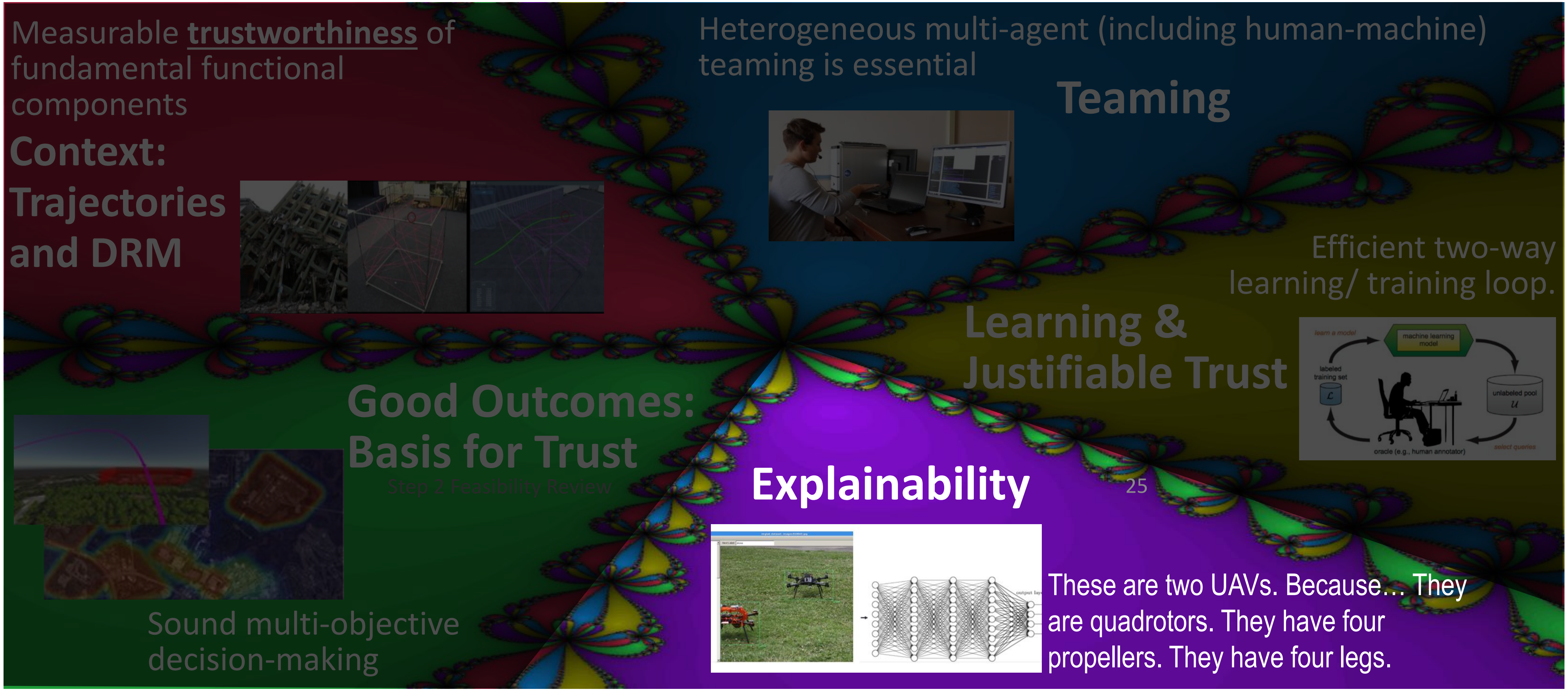
x* = argmin F(x), where F(x) = $\sum w_i \, f_i(x)$ is a decision objective, local or global, $f_i(x)$ are known; e.g., {distance, fuel, delays,…}

E.g., Use interactive multiobjective optimization and inverse reinforcement learning to ID common values of weights


Autonomous
Multi-Modal
Multi-Agent
Search & Rescue

Machine Learning (ML) and Explainable AI (xAI)

Multidisciplinary Components of
Complex Multi-agent Systems



Motivation

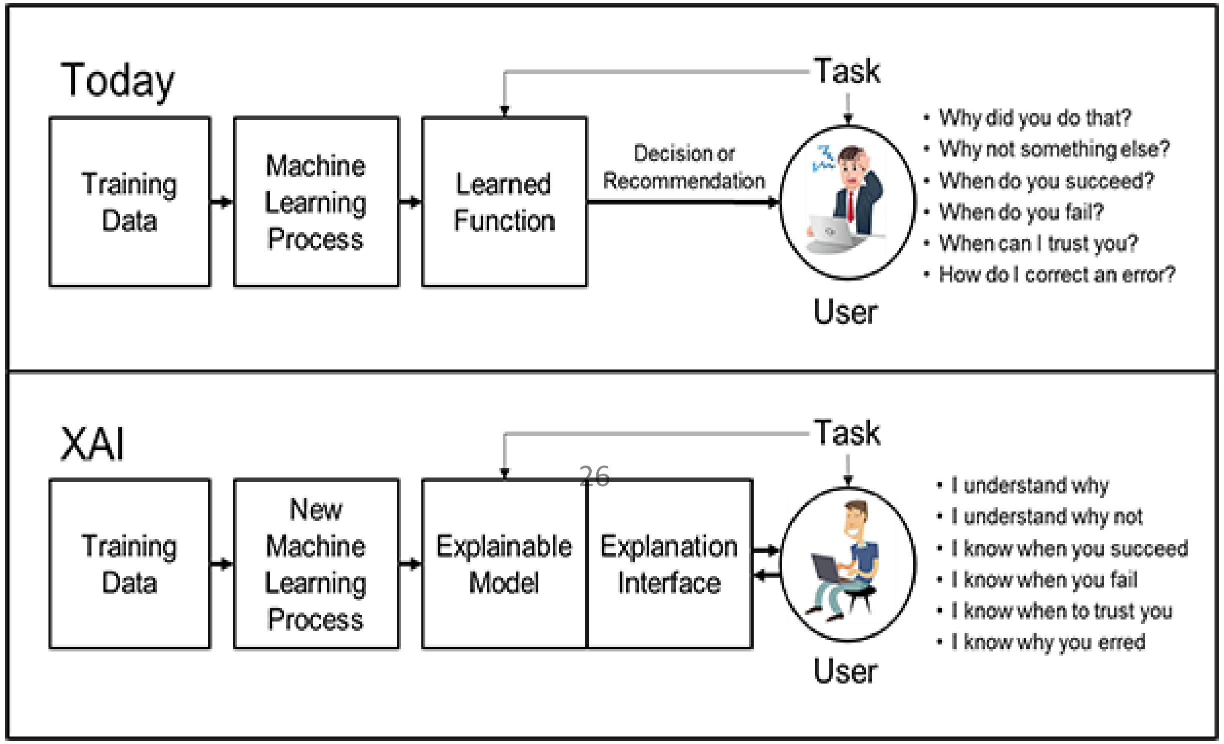
- ML is fragile. Why ML in safety-critical environments?
  - Situational awareness with sensors that require interpretation (perception) relies on ML
  - ML can also serve in an advisory capacity

- Why XAI?

↑ T&T

  – Understand decision-making processes; debug; train; a posteriori analysis
  – Humans prefer decisions with explanations in accessible terms

DARPA's XAI research



https://www.darpa.mil/program/explainable-artificial-intelligence

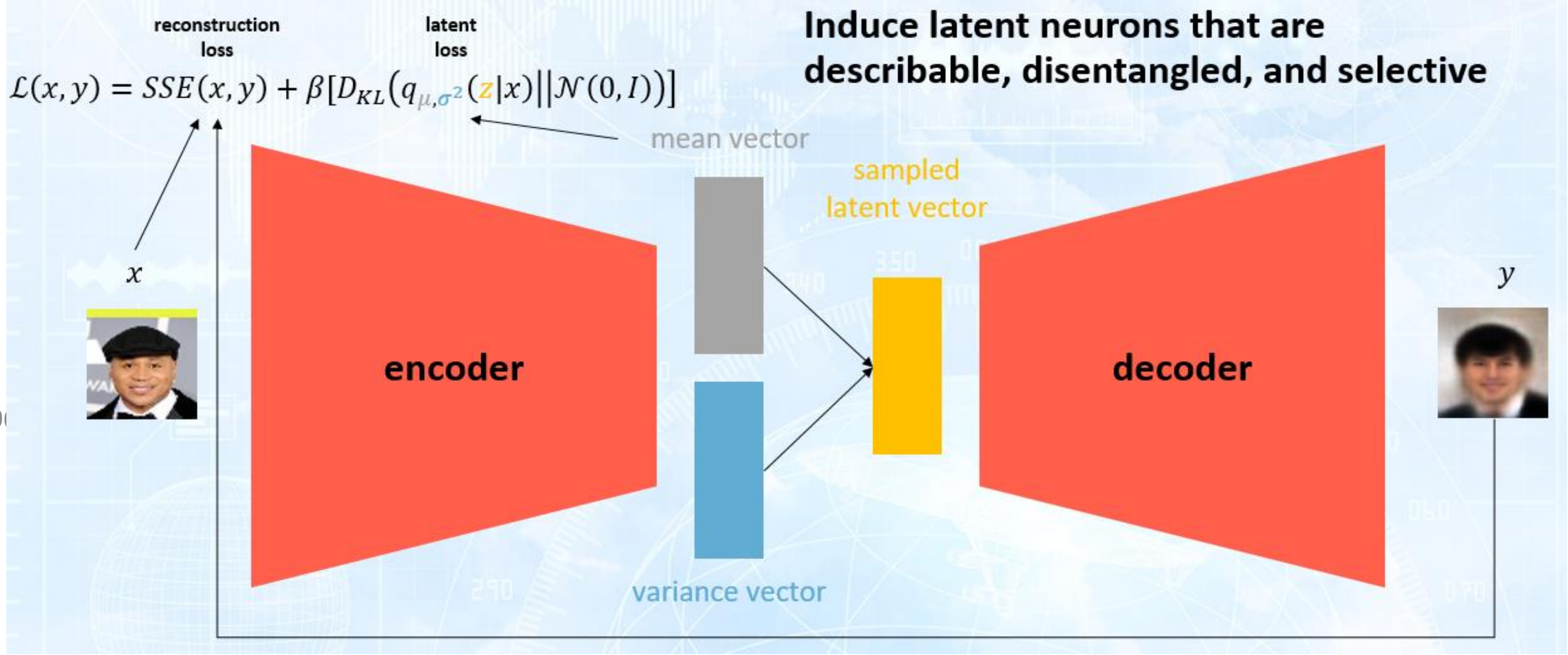
Step 2 Readability Review

Explaining ML with Variational Autoencoders (VAE)

Loc Tran

- <u>Idea/Concept</u>: Is there a network neuron with high selectivity to a feature that can be clearly described (e.g., is there a 'red shirt' neuron?) and can we overcome entanglement?

The $\boldsymbol{\beta}$-Variational Autoencoder
places weight on disentanglement



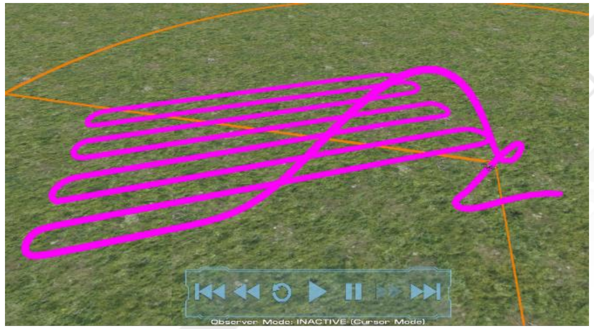
- Autoencoders are neural networks that learn representation of data
- Autoencoders compress and reconstruct input images
- $\boldsymbol{\beta}$-VAE goal: learn disentangled latent parameters from unsupervised data
- MOISTURIZE Tool: developed in house to explore implementation of VAE

Search and Rescue (SAR) Mission

Deploy UAV

Search Phase

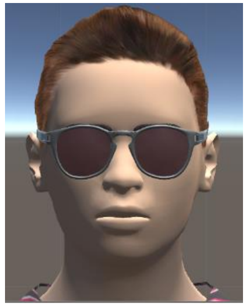
Search Path



Detect Person

Move in

closer



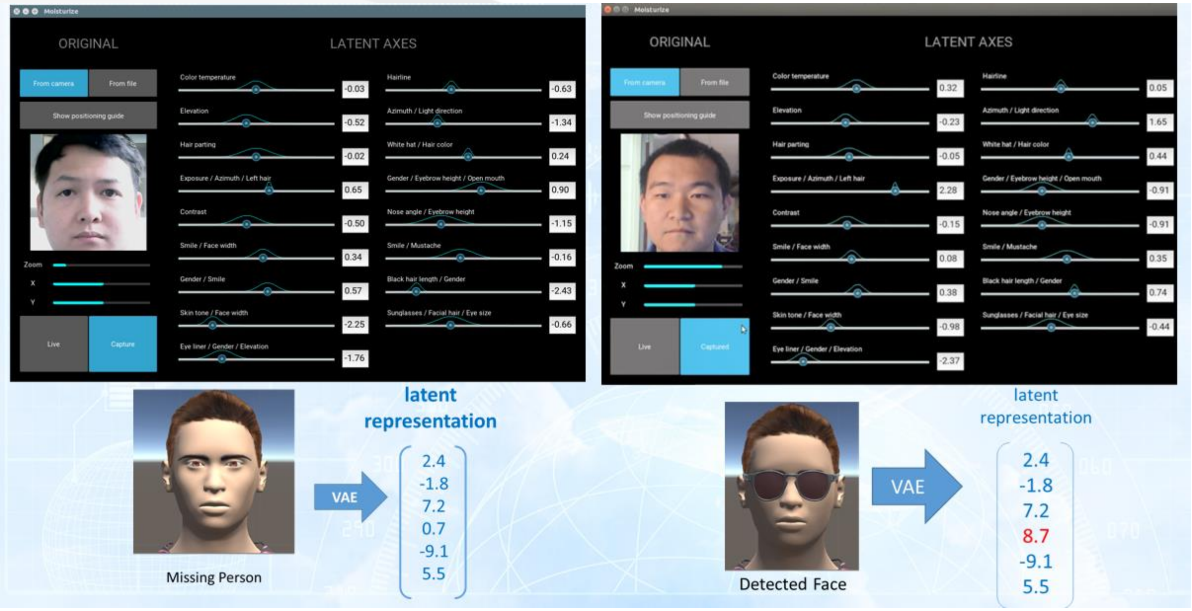
Detect Face

VAE

2.4
-1.8
7.2
8.7
-9.1
5.5

latent
representation

Detect Phase



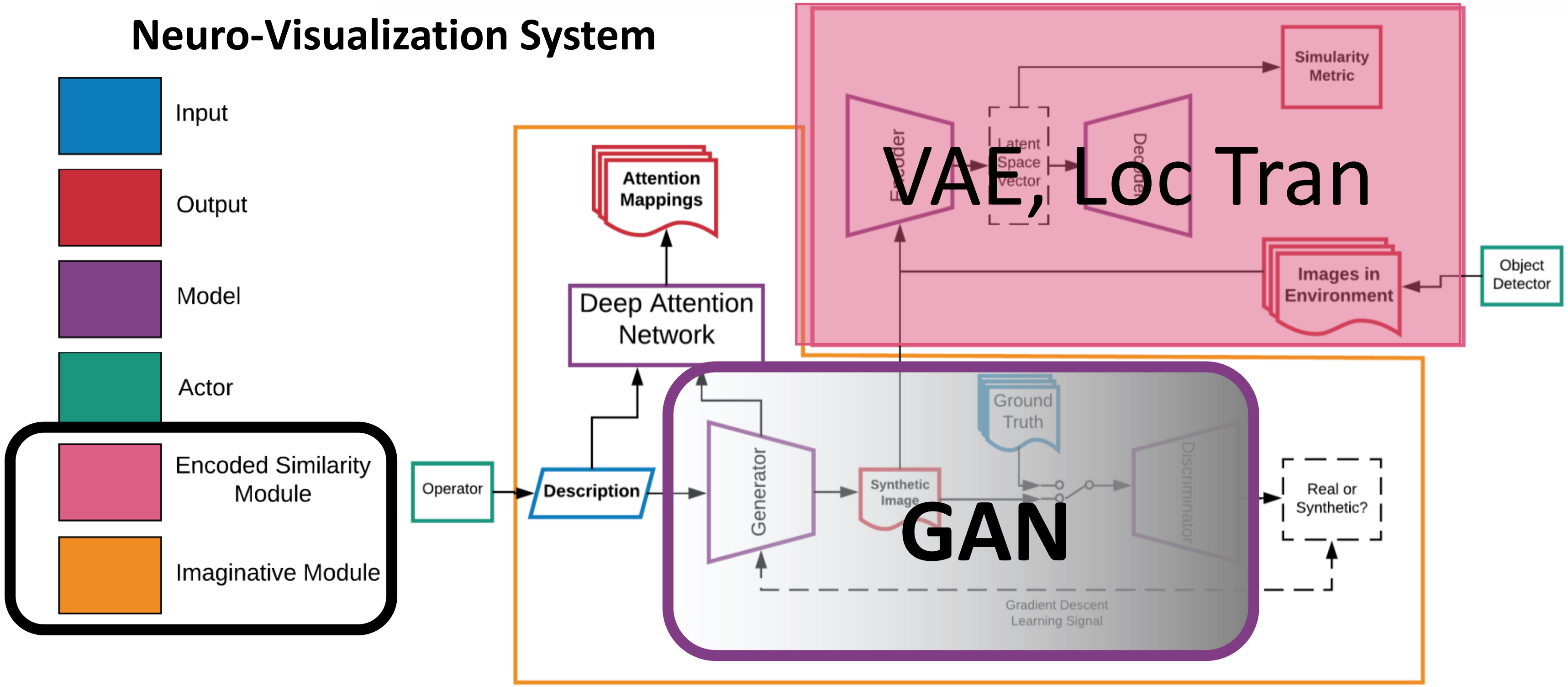
Findings:

- VAE are a feasible tool for representing human interpretable features from complex data sets
- VAE do not produce completely disentangled representations
- Suitable as additional info or advisor; not for direct safety-critical action

Deep Learning for Neuro-visualization
James Ecker

- Idea/Concept: Describe an object of interest via neuro-visualization: tell a machine **what it is looking for**

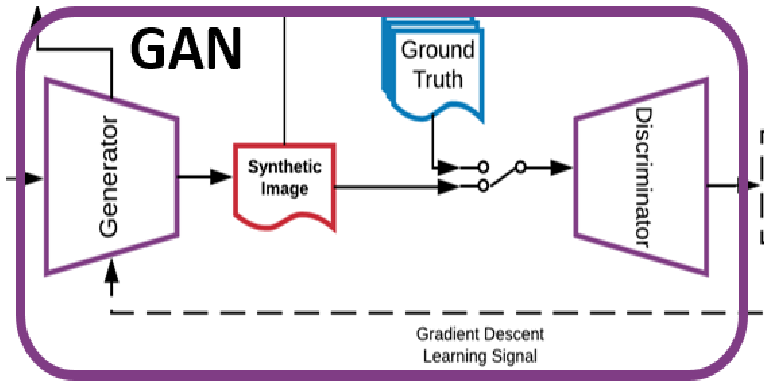
Neuro-Visualization System



Imaginative Module Generates pixel-space representation of object described in natural language

Encoded Similarity Module Converts target and observed image pairs into respective feature vectors via a pre-trained model

Generative Adversarial Network



- **Generator:** Generates synthetic data to try to fool the Discriminator
  Discriminator: Classifies whether an instance of data is "real" or not
  Minimax Game: Generator and Discriminator engage in game that ends when the discriminator can get no better than chance (50% accuracy)

Neuro-visualization: Examples of Findings

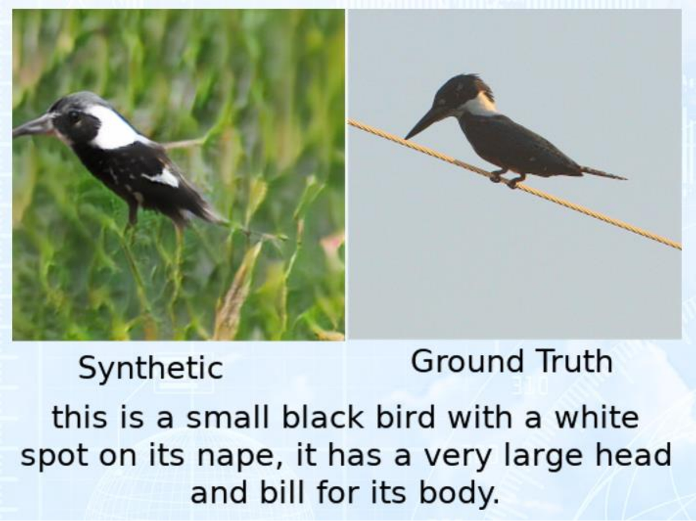
Birds

Good results rely on description context and localization of object in the training data

Humans

COCO dataset: Descriptions contain information holistic to the image, with low object specificity

Machine Imagination



Reconstruction from ground truth shows highly accurate but imperfect recall

How do we tell a machine what to look for?

And how do we know (trust!) that it understands?

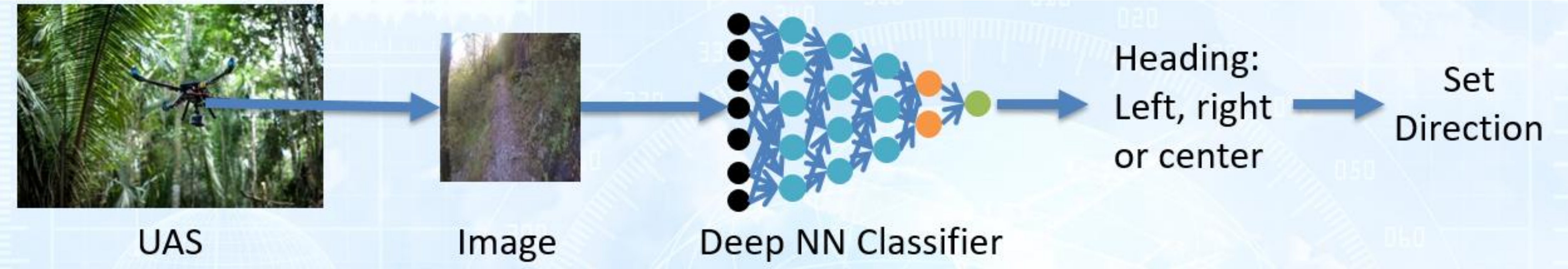
Findings:

- Given properly configured data, this system can reliably generate explainable images of the subject classes
- Generated images, if not accurate as a whole, possess a collection of features meaningful to the algorithm
- Shows promise for operational xAI in the context of SAR; feasible as advisor to decision-maker

Improving Trust in Machine Perception through Explanations

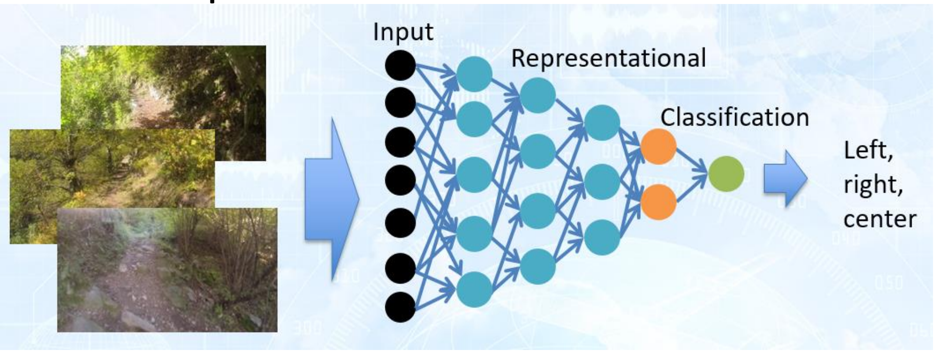
Adrian Agogino, Ritchie Lee, Dimitra Giannakopoulou

- Idea/Concept: Use explanation by example, nearest neighbors (KNN), in latent space for transparent and understandable decision making
- KNN is a viable but more intuitive alternative to standard decision-making for deep neural networks
- Test Domain: Neural Networks (NN) for Forest Trail Image Classification
  - UAS navigates with front facing camera; determines direction to follow trail

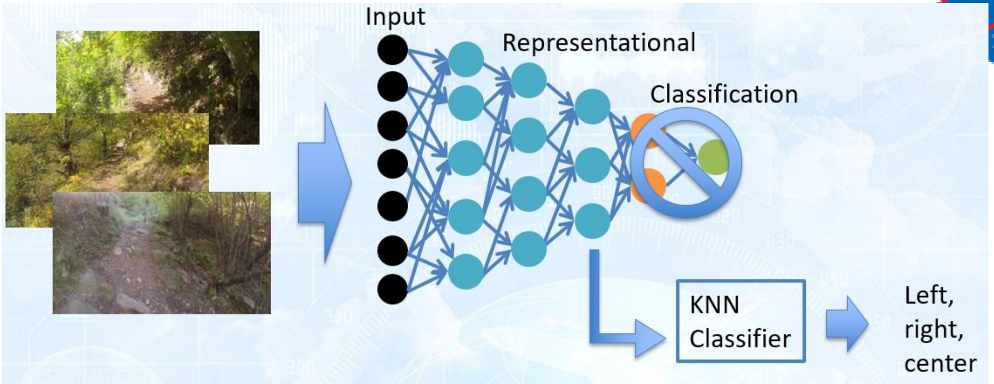


KNN, cont.

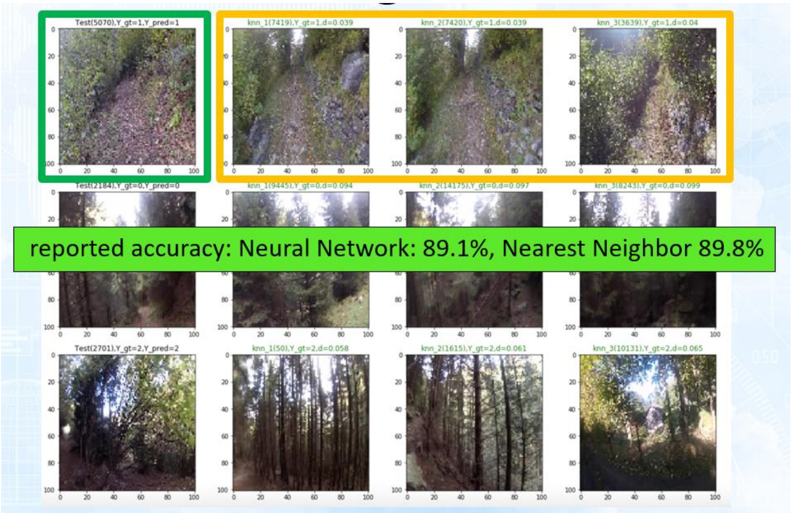
Deep NN Classification



K-Nearest Neighbor



3-Nearest Neighbor Classifier (cosine similarity)



Time-Series Images

KNN identified unknown problem in time series data

Corrections were made to give more realistic performance measure

Findings:

- Performance similar to SOA deep NN
- Nearest neighbors help detect errors in training
- Visualization of neighbors indicates quality of distance functions
- Decisions are transparent
- Explanations are intuitive since they relate decisions to training data

Trustworthiness of ML-enabled Safety-critical Systems

Alwyn Goodloe

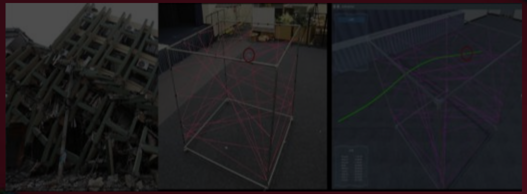
- **Findings:**
  - At current SOA, it may not be possible to write specifications of the function correctness for SOA ML systems. This makes them not amenable to traditional software verification techniques.
  - Basic research remains to be done on how to verify functional correctness of ML enabled systems.
  - A number of researchers are investigating verifying correctness properties of neural networks.
  - For the foreseeable future, runtime verification will have to be used extensively. However, runtime verification faces the same challenge as formal verification approaches: the need to specify the property being verified.
  - Perception poses a special problem because properties amenable to formalization are highly specialized and have yet to be invented for perception.

Toward T&T Design and Operations

Multidisciplinary Components of Complex Multi-agent Systems

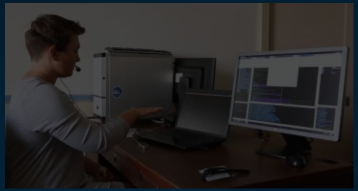
Measurable **trustworthiness** of fundamental functional components

Context: Trajectories and DRM



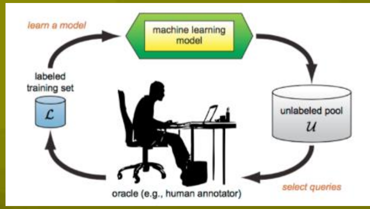
Heterogeneous multi-agent (including human-machine) teaming is essential

Teaming

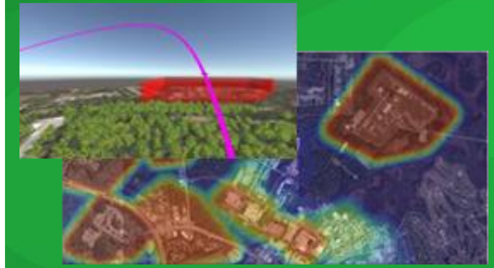


Efficient two-way learning/ training loop.

Learning & Justifiable Trust



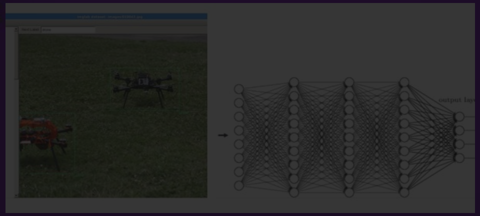
Good Outcomes: Basis for Trust



Sound multi-objective decision-making

Explainability

34



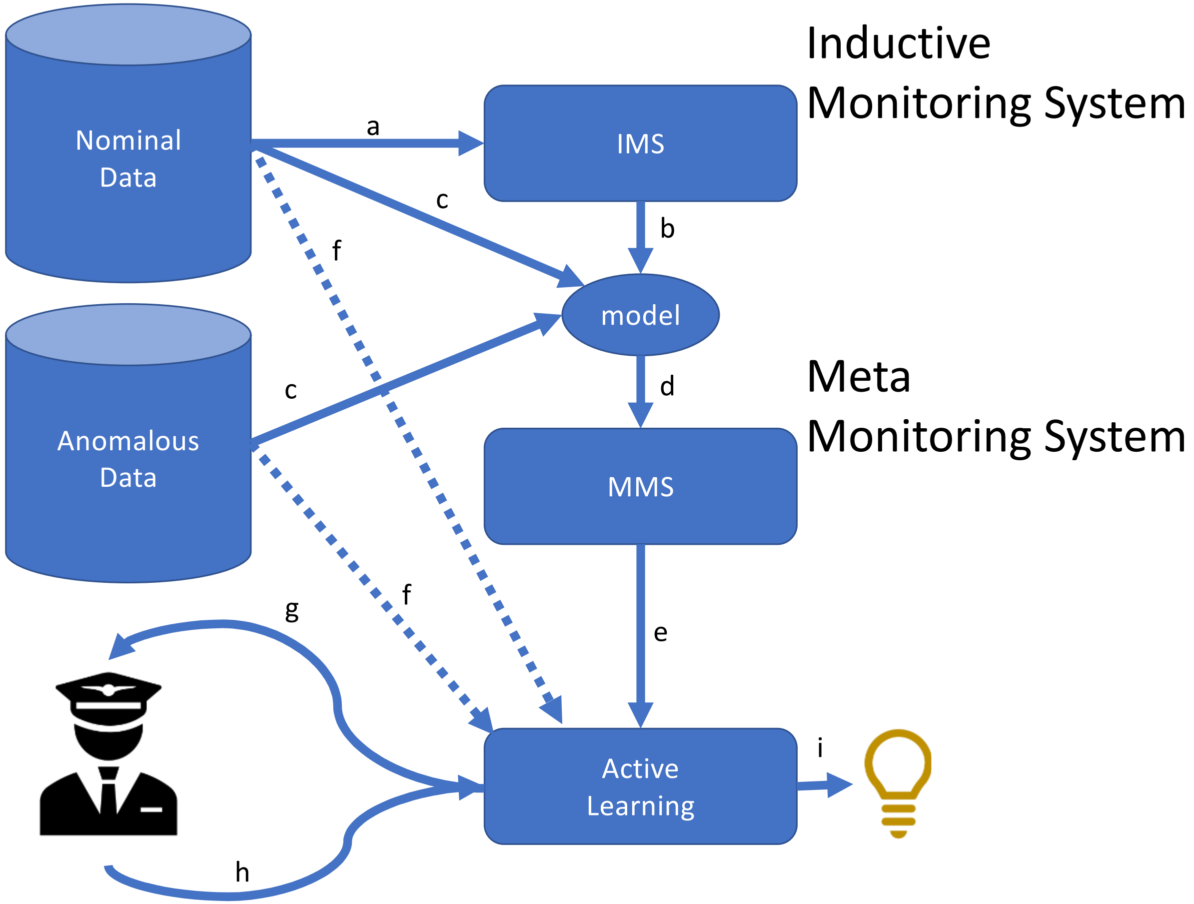
These are two UAVs. Because… They are quadrotors. They have four propellers. They have four legs.

Incorporating Human Knowledge in Autonomous Systems (AS) through ML

Nikunj Oza, Kevin Bradner, David Iverson, Adwait Sahasrabhojanee, Shawn Wolfe

- Idea/Concept: Facilitate trust in autonomous systems by using domain expert knowledge to identify anomalies and their precursors. Explain anomalies during operations.

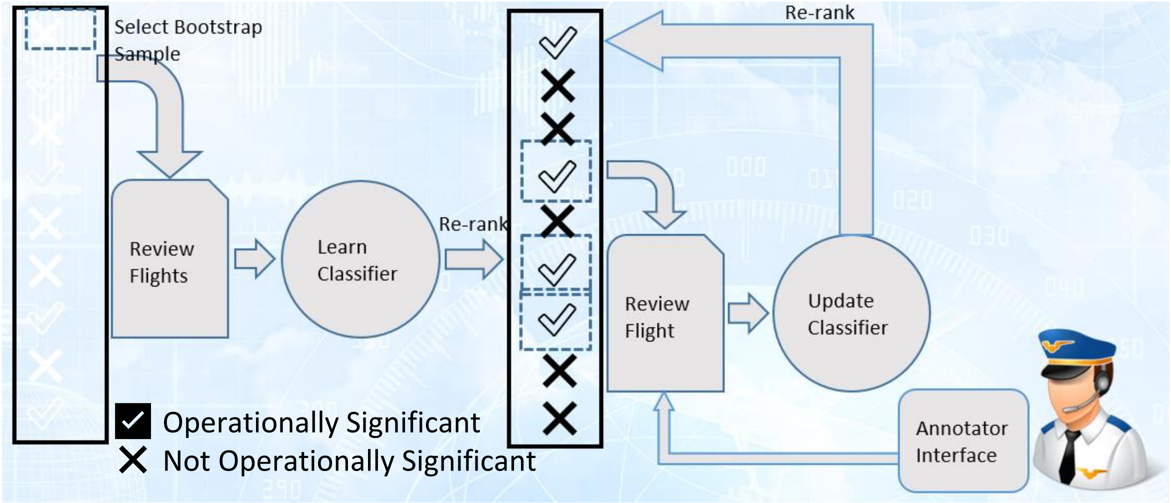
Anomaly Detection Pipeline



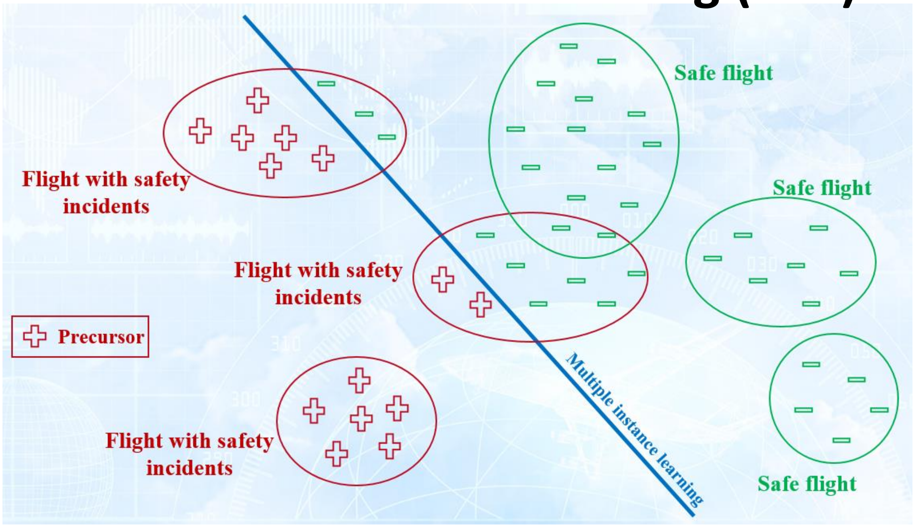
- IMS deviation scores need interpretation
- MMS post-processes IMS scores
- MMS score evaluates probability that each observation was generated from an off-nominal system

Anomaly Detection, cont.

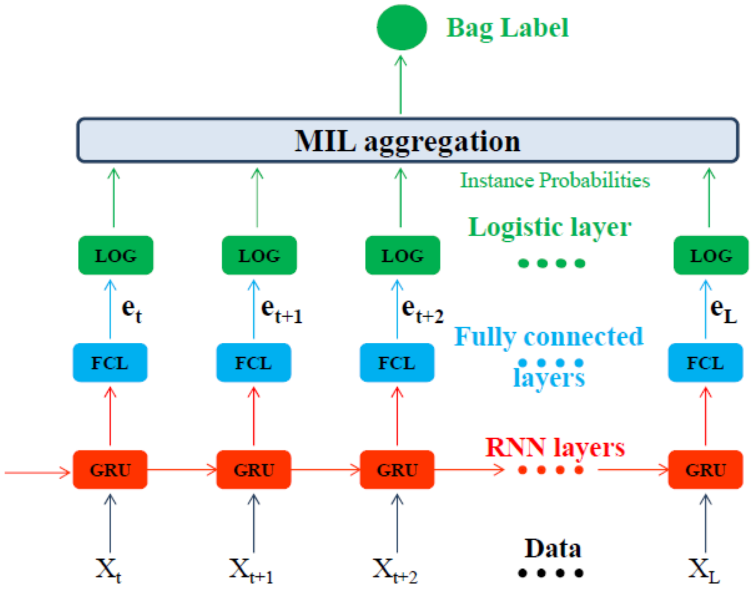
Active Learning



Multi-instance Learning (MIL)



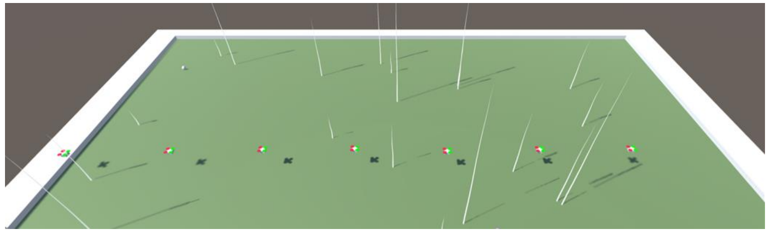
Deep Temporal MIL (DT-MIL)



Testing

- 100 simulations of each failure type (rotor failure, tree collisions)
- Varied search area, number of drones, operating characteristics
- Train and test within scenarios, across scenarios

Findings:

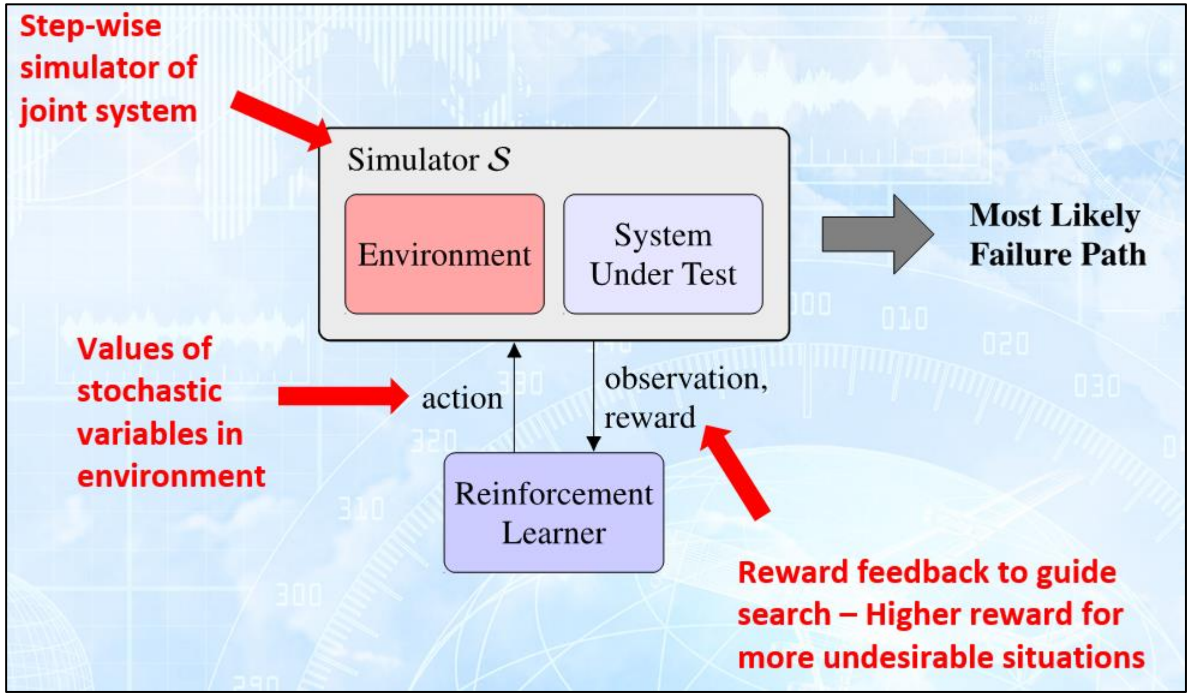


- Successful detection of operationally significant vs. non-operationally significant anomalies
- Successful identification of precursors to anomalies

Adaptive Stress Testing (AST) of Trajectory Planning Systems

Ritchie Lee, Javier Puig-Navarro, Adrian Agogino,
Dimitra Giannakopoulou, Ole Mengshoel, Mykel Kochenderfer, Danette Allen

- Idea/Concept: Validation of Trajectory Planners (TP) using Adaptive Stress Testing (AST) to increase confidence in TPs



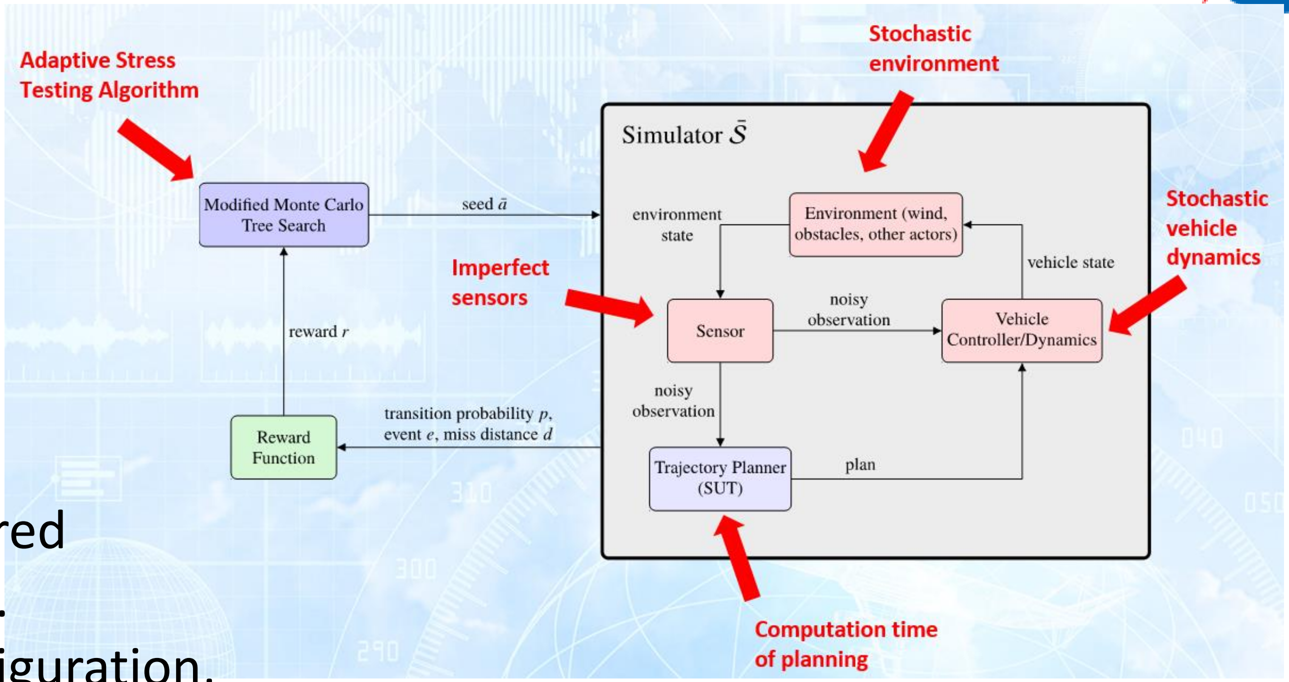
SOA (Monte-Carlo) does not scale

AST

- Formulates finding the **most likely failure path** as a **sequential decision-making problem**
- Apply reinforcement learning algorithms for finding failures
- Successful applications in Aircraft collision avoidance systems (ACAS X) and self-driving cars

AST: Application to Wire Maze Scenario

Testing Architecture



Experiments:

- Stressed for collision with obstacles; considered planning failures, implementation issues, etc.
- Testing dimensions: Sensor noise, initial configuration, replanning condition, computation time, dynamics noise

Results:

- 1600 total scenarios searched
- 386 successfully reached goal
- 19 collisions
- 1195 terminated in planning failure

Findings:

- AST can be extended and scaled to trajectory planning
- AST can run unanticipated scenarios and find a variety of errors
- Helped understand the residual risk, failure modes of the system
- Increased understanding of system’s trustworthiness

Leveraging Fault-Tolerance Concepts as a Basis for Distributed Trust

Paul S. Miner

- Idea/Concept: Justifiable *trust* is derived from a collection of *trustworthy monitors* that are coordinated using *trustworthy consensus mechanisms*
- Approach (work in progress): The working conjecture is that we can develop a workable model of trust by identifying/defining monitors and consensus mechanisms and determining how to ensure that the monitors and consensus mechanisms are sufficiently trustworthy for the level of trust required of the overall system
- Example of included attributes: Availability, Reliability, Safety, Confidentiality, Integrity, Maintainability; Relevant Knowledge; these are reasonably mature concepts but difficult to establish
- Hypothesis and Ongoing work: Well-established concepts from Dependable Computing apply to autonomous system and must be focused in specific contexts

“Mind” problems

Ab initio design of future airspace

IRAD project of managing airspace complexity

Settings

A critical aspect of uncertainty in the operation of complex systems: the ability of agents to arrive at satisfactory decisions and the attendant actions in time, as a function of problem complexity

Statistical Estimates from the Automobile Domain

- How many miles of driving would it take to demonstrate autonomous vehicle reliability? (Kaira & Paddock)

Miles/Years to be driven…	1.09 fatalities per 100 million miles	77 reported injuries per 100 million miles	190 reported crashes per 100 million miles
To demonstrate with 95% confidence that the failure rate is at most…	275 million miles (12.5 years)	3.9 million miles (2 months)	1.6 million miles (1 month)
To demonstrate with 95% confidence their failure rate to within 20% of the true rate of…	8.8 billion miles (400 years)	125 million miles (5.7 years)	51 million miles (2.3 years)
To demonstrate with 95% confidence that their failure rate is 20% better than the human driver failure rate of…	11 billion miles (500 years)	161 million miles (7.3 years)	66 million miles (3 years)

Complexity

- Reducing uncertainty to achieve manageable operation is always done via bounding problem complexity

- Current airspace control:
  - Static complexity bounds
  - Related to cognitive capacity of human controller

- Goal: Bound complexity in a dynamic and scalable way amenable to computational decision-making in human-machine and autonomous machine systems

- Want:
  - Keep the system sufficiently simple
  - Represent complexity in computable, actionable form
  - Detect approaches to unacceptable complexity
  - Reconfigure system to forestall transition to unacceptable complexity
  - Reconfigure system once increased complexity is resolved

Proposed Measure of Complexity (NMA)

- Tractability, with a look ahead, of the solution problem on a time budget, as a function of the external and internal parameters. For air traffic:
  - External parameters, e.g.: density and heterogeneity of the relevant airspace volume
  - Internal parameters, e.g.: physical properties of the aircraft and computational properties of decision-making algorithms

+

- The quality of solutions of the decision-making problem measured in terms of constraint satisfaction, optimality, and robustness. For air traffic:
  - One measure of robustness: robust solutions live in the regions of space that allow for many alternative solutions (e.g., flexibility preservation, Idris et al.).

Conceptual Approach

Given:

An agent and a goal

Initialize:

Set time t=0

Set initial sampling time interval Δt

Set initial look-ahead time T

Set initial look-ahead sampling time interval $\Delta\tau$

Select initial problem-solving algorithm P

Select environmental complexity parameters C

Select initial environmental complexity model M

Select transition criteria

Select stopping criteria

Conceptual Approach, cont.

```
Do until (stopping criteria are satisfied)
Acquire and assess complexity parameters C at time t
Set τ = t
   Do while (τ ≤ t + T)
              Estimate look-ahead complexity parameters at time τ
              τ = τ + Δτ
   End do
   Input complexity parameter array to complexity model M; assess approach to phase transition
If (approach to transition detected) then
              Update Δt , Δτ , T
   Reconfigure operations toward goal
   Else
              Assess system's performance slack
              If (slack)
                  Reconfigure operations toward goal
              Else
                 Continue present operations toward goal
              End if
   End if
   t = t + Δt
End do
```

MAGE (Monitor, Anticipate, Guide, Evolve) for Air Traffic

- Complexity model construction:

Number of aircraft
Aircraft velocities
Decision-making scheme → ML or Other Data-Fitting Model) → Estimate of solution time
Estimate of solution quality

- Reconfiguration: change in the decision problem objectives, constraints, and variables
  - Directed modification
  - Autonomous, distributed modification with emergent outcomes
  - Hybrid directed-autonomous

  - Example: reduce weight of the delay objective in high-risk, dense environment

- Decision-making strategies:
  - Solution strategy (e.g., optimization) affects the outcome of actionable complexity prediction.
  - As new capabilities arise, complexity models must be re-calibrated

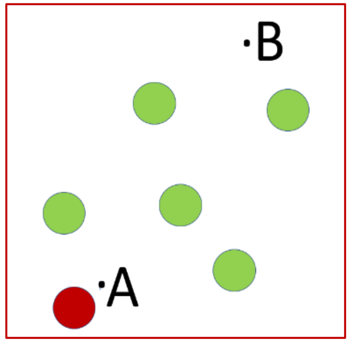
Initial Numerical Tests of Tractability Prediction

- Shortest Path Finding via Visibility Graph

# obstacles	# scenarios	Avg. # solutions	Avg. time to solution (sec)	# test scenarios	Avg. # test solutions	Avg. % time prediction Δ
4	100	97	0.86	25	24	0.02
5	100	89	1.31	25	19	0.07
10	100	34	324.33	25	7	2.03
20	100	2	4800.00	25	Not found	N/A

- Minimization of Deviation from Optimal Path

# obstacles	# scenarios	Avg. # solutions	Avg. time to solution (sec)	# test scenarios	Avg. # test solutions	Avg. % time prediction Δ
4	100	94	0.38	25	25	0.01
5	100	100	0.37	25	23	0.01
10	100	83	11.60	25	24	0.03
20	100	86	64.52	25	19	0.05



- Detect trends in computational tractability of simple decision problem formulations, using examples of two formulations and two decision-making schemes.
- Results to date indicate that further development of the planned complexity models is justified.

Experimental Set-up

- ATMLG (Air Traffic Monotonic Lagrangian Grid)
  - **Initialize system**
    - N = number of aircraft
    - System aircraft interaction distance (10 mi)
    - Set time t_i = start time_i (can be 0 or chosen randomly from a set) for i=1,…,N
  - **Do ∀ 10 sec until End**
    1. ID all aircraft in conflict (within interaction distance)
    2. For ∀ aircraft in conflict, select: {x_1 = Δ heading±45°, x_2= Δ speed±10 kt, x_3= Δ altitude±1000 ft}
    3. For ∀ pair of aircraft in conflict, formulate a constraint:
       1. Compute t of closest approach for 2 aircraft based on current position, heading and speed (regardless of altitude)
       2. Compute separation distance between the 2 aircraft at t of closest approach
       3. Compute a projected altitude factor

$$f_1 = 1 - \frac{(alt_1 - alt_2)}{Alt_{sep}} \qquad f_2 = 1 - \frac{(alt_2 - alt_1)}{Alt_{sep}} \qquad g = \left(1 - \frac{distance_{closest}}{distance_{allowed}}\right) * f_1 * f_2$$

    4. Solve minimize $\sum \left(\frac{x_i}{scale_i}\right)^2$, subject to $g \leq 0$
  - **End Do**

Sample of Results

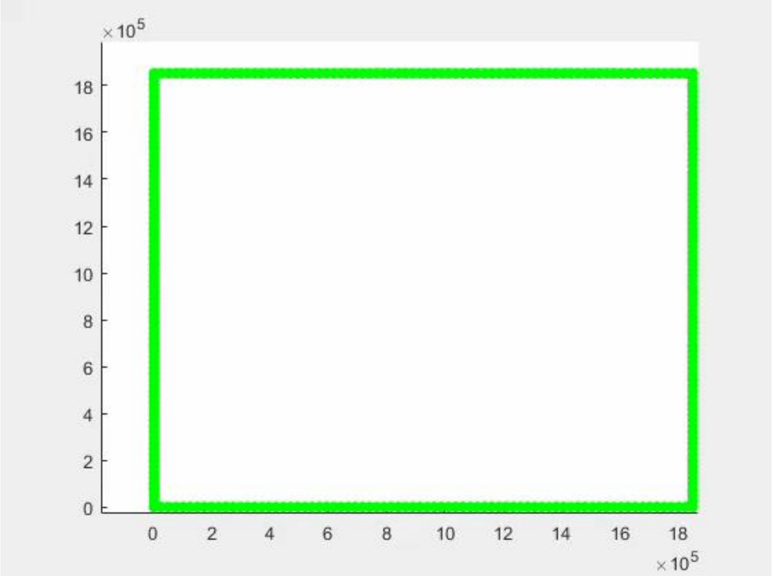
- Data Sample

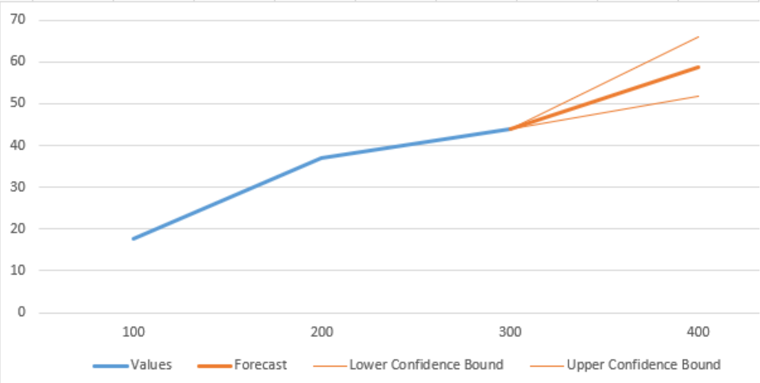
Training set

Validation set

Different optimizer

1	All planes flying at a constant altitude of 30,000 feet.									
2	All planes flying at a constant airspeed of 400 knots.									
3	Check distance is 10 nautical miles and intrusion distance is 5 nautical miles.									
4	Circle radius is 1000 nautical miles.									
5	All planes start at time=0.									
6										
7–8	Optimizer name	number of planes	initial separation	trial number	number of conflicts	number of course changes	closest approach	execution time	optimizer total time	optimizer worst case time
9	KSOPT	100	62.83	1	118	89	5.00	19.187500	0.026000	0.008000
10		100	62.83	2	118	103	5.00	16.609375	0.055000	0.009000
11		100	62.83	3	98	70	4.50	17.171875	0.012000	0.008000
12		100	62.83	4	134	125	5.00	17.546875	0.038000	0.001000
13		100	62.83	5	126	111	5.00	17.656250	0.016000	0.001000
14		200	31.42	1	500	495	2.50	29.781250	0.261000	0.010000
15		200	31.42	2	482	545	1.70	29.031250	0.193000	0.020000
16		200	31.42	3	410	400	3.60	27.500000	0.232000	0.021000
17		200	31.42	4	508	423	4.60	28.281250	0.215000	0.021000
18		200	31.42	5	446	415	4.20	27.406250	0.259000	0.010000
19		300	20.94	1	1066	1034	1.90	42.968750	2.955000	0.402000
20		300	20.94	2	1064	1083	1.50	44.015625	3.389000	0.419000
21		300	20.94	3	1148	1049	2.60	11.859375	2.359000	0.203000
22		300	20.94	4	1052	1004	2.50	12.968750	3.735000	0.923000
23		300	20.94	5	1072	1058	0.90	13.890625	3.677000	0.616000
24		400	15.71	1	1906	1865	1.10	33.375000	22.192000	2.733000
25		400	15.71	2	1972	1892	0.70	47.406250	34.176000	7.738000
26		400	15.71	3	1910	1781	0.90	30.625000	19.172000	3.087000
27		400	15.71	4	2008	2026	1.30	39.984375	27.847000	4.613000
28		400	15.71	5	1984	1944	1.70	38.609375	26.579000	3.765000
29	NLOPT_MMA	100	62.83	1	84	69	5.00	36.546875	31.311000	12.550000
30		100	62.83	2	100	65	4.30	65.031250	59.835000	12.488000
31		100	62.83	3	82	44	3.60	77.453125	70.474000	11.719000
32		100	62.83	4	118	67	3.10	17.906250	11.362000	3.866000
33		100	62.83	5	114	125	4.60	81.093750	75.586000	10.797000

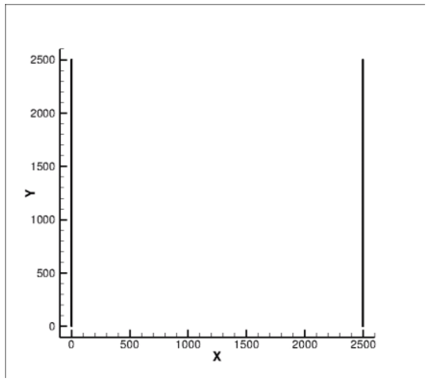




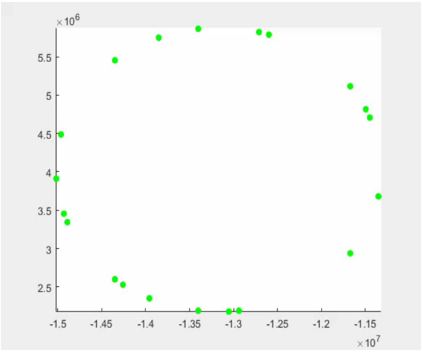
Timeline	Values	Forecast	Lower Confidence Bound	Upper Confidence Bound
100	17.63438			
200	36.99375			
300	44	44	44.00	44.00
400		58.77883	51.65	65.91
		actual 38.		

Open Question

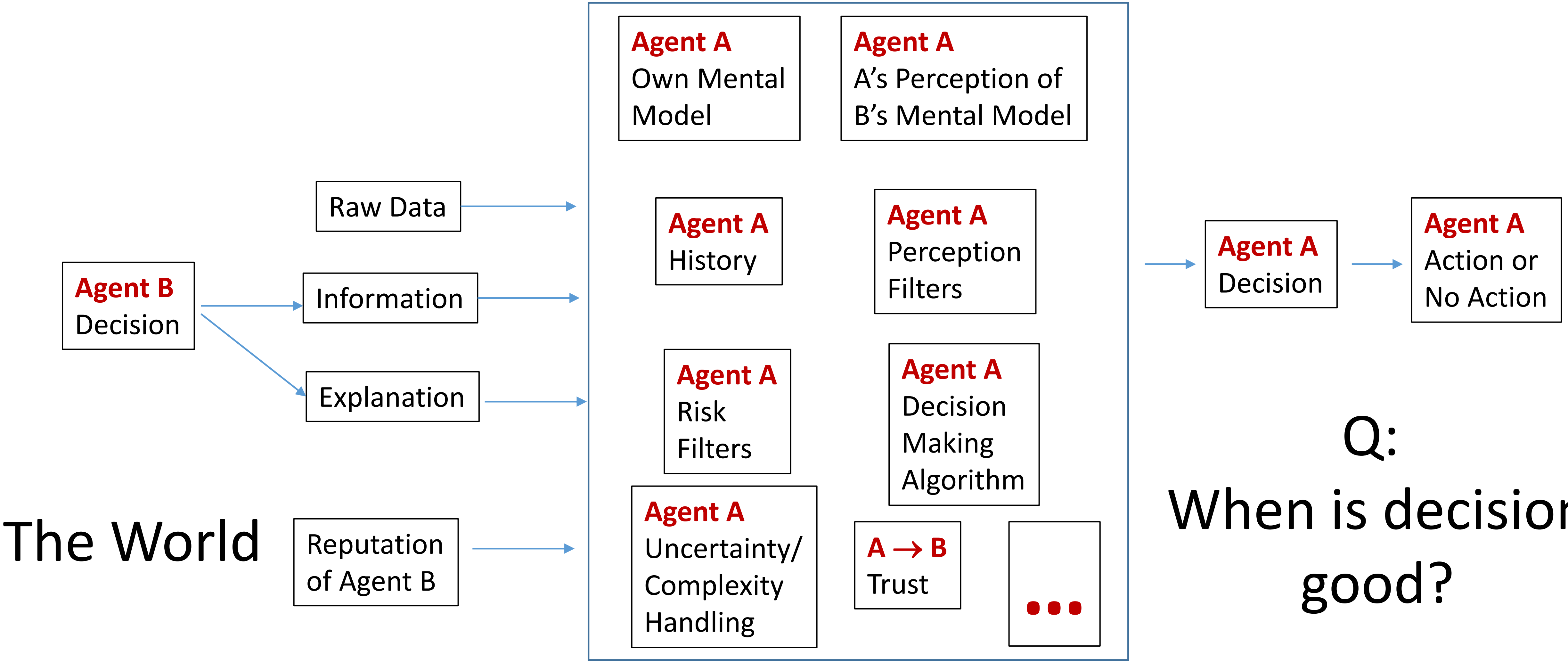
- Problem
  - Suppose information is available to all and perfect
  - Suppose all participants use the same optimization algorithm
  - Satisfactory solutions not guaranteed
- Current setup: Locally centralized
- To do:
  - Find the minimum necessary (and sufficient?) commonality in heterogeneous decision-making to ensure viability of any architecture



Information and decision-making assumptions



Complexity of Agent's Decision Making



Q:
When is decision good?

“Body” problem

Survivability in collisions

Effects of Density on Safety

- Probability of collisions as a function of density
  - Not enough data to formulate a historical model
  - Gas particle models give an estimate (e.g., Alexander 1970)
- Consider randomly distributed aircraft

  r = a given interaction distance

  S = distance traveled by an aircraft

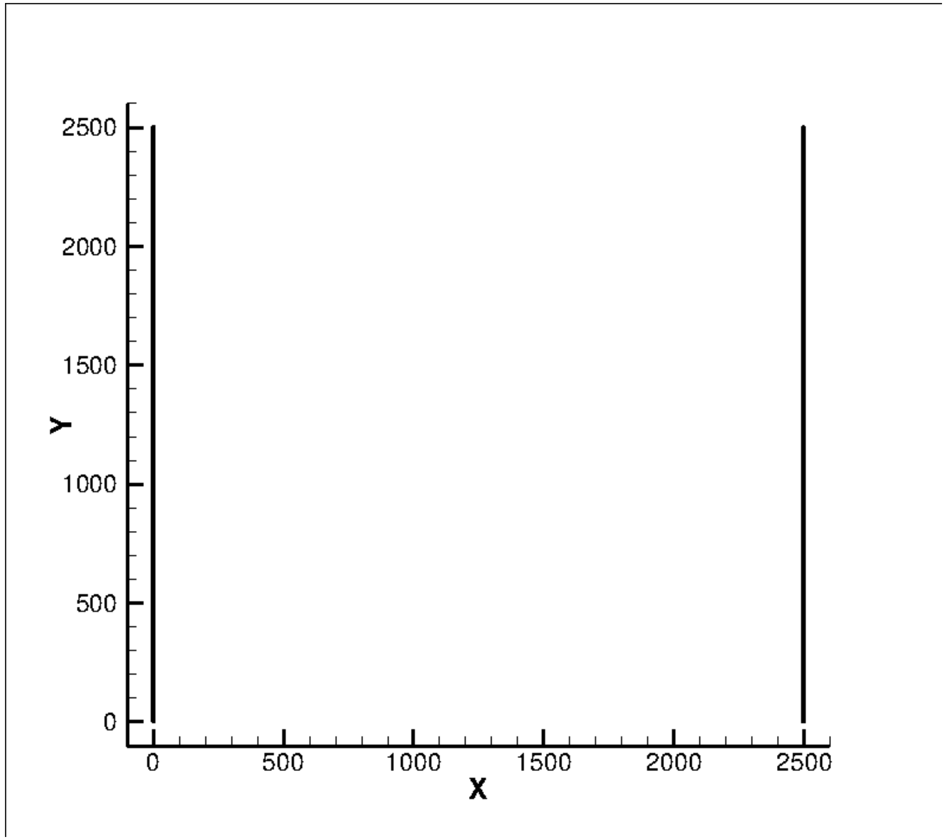
  N = average number of aircraft/unit land area

  L = number of altitude layers; aircraft equally divided among layers

  $P = e^{-\frac{2rSN}{L}}$ = the probability that an aircraft moving horizontally a distance S will not come within distance r of another aircraft in the same layer
- Example: for r = 1 mi, L = 10, velocity = 150 mi/h, and 3000 mi^2
  - ***N* = 0.01 aircraft per mi^2**: hours for 2-body interaction and 1000 hours for 3-body interactions
  - ***N* = 0.1 aircraft per mi^2**: evasive maneuvers required every 20 minutes
  - ***N* = 1.0 aircraft per mi^2**: nearly continuous maneuvering
- Supported by frequency of near misses in dense airspace near airports
- Airspace structure is an option, but not in door-to-door on-demand mobility (ODM)

The Biggest Problem: Heterogeneity

- With shared rules, great densities can be accommodated (joint work with NRL)

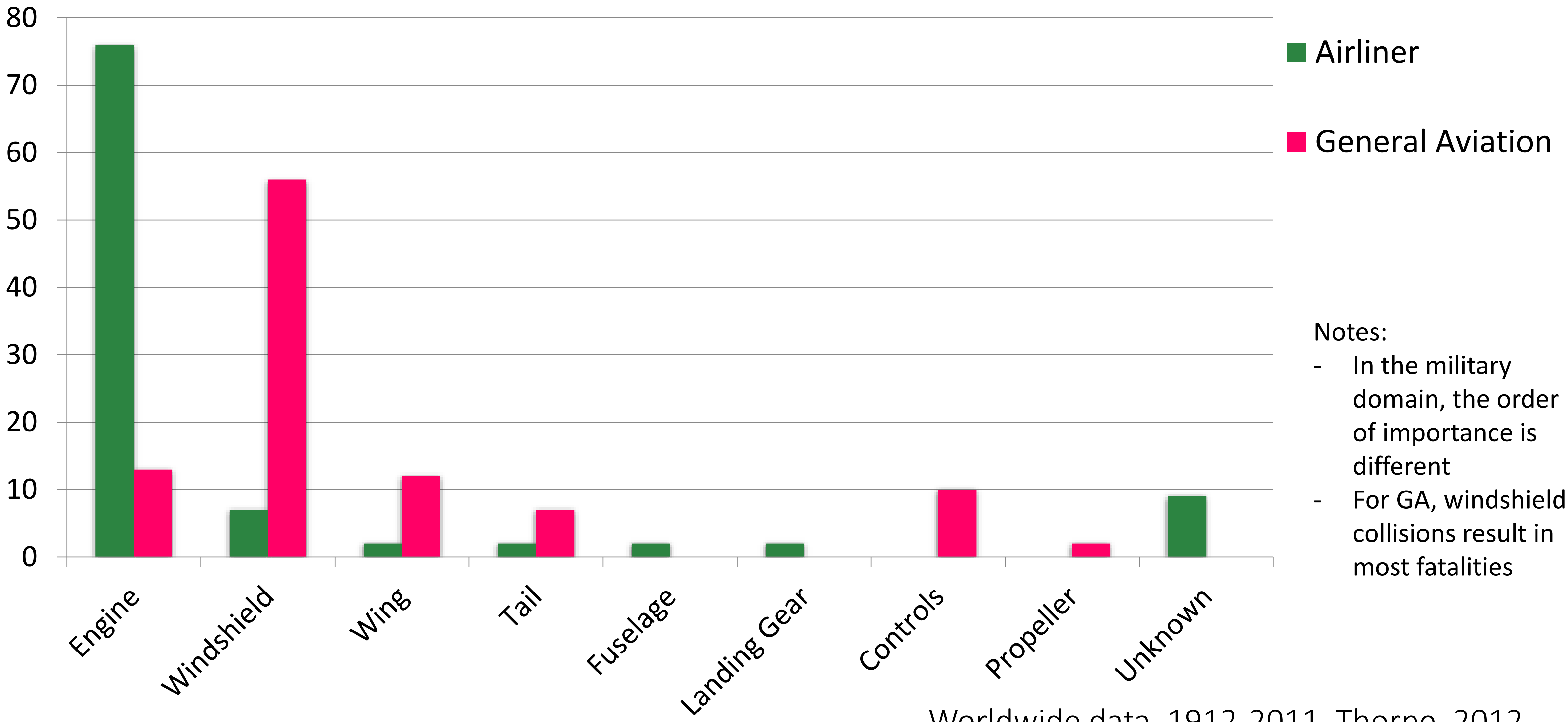


- With heterogeneity, e.g.:
  - Interference with CA firefighting efforts
  - “Drone Close Calls, Sightings by Airliners up Fivefold” (3/28/2016, ATCA News)
  - “Miracle on the Hudson”: 1.5 sec between “bird!” and “collision!”

On Collision Damage in the Civil Domain

- A recent activity, with a lot of unknowns (e.g., Virginia Tech CRASH lab FEM modeling of collisions, Bayandor et al.; Radi 2013)
- Use wildlife collisions as a model for frequency of incidence with an object
- Rough estimates for where to focus, to minimize damage
- Cannot completely extrapolate from birds to UAV

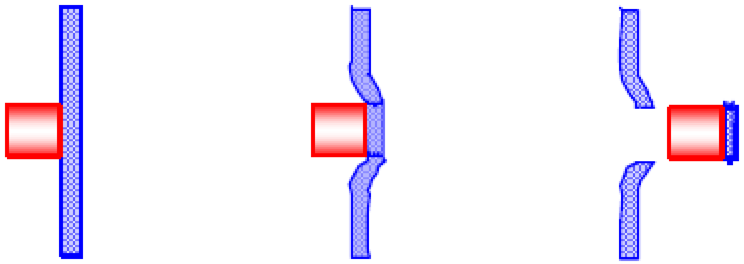
Location of Bird Strikes (%)

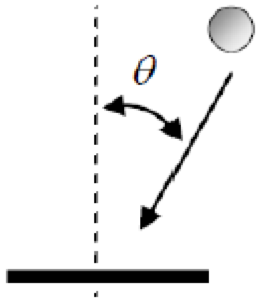


Notes:
- In the military domain, the order of importance is different
- For GA, windshield collisions result in most fatalities

Worldwide data, 1912-2011, Thorpe, 2012

Impact Damage





- Studies, e.g.,
  - Bird impacts, based on historical data (e.g., Thorpe 2012)
  - Studies of UAV impact damage, using the FAA penetration equation $V_{50} = \sqrt{\frac{2LCSt^2}{m\cos^2\theta}}$ = the ballistic limit = velocity required to make a hole in a sheet of metal. Here m is the mass of the projectile, θ is the angle of the impact, C_S is a material property constant, L is the perimeter of the presented area of the projectile, t is the thickness of the metal sheet (e.g., Radi 2013)
  - Many assumptions; large uncertainties, but overall trends relevant
- GA aircraft: Penetration likely at cruise speeds, regardless of UAV size; and likely for large UAV during approach

An Approach to Problem Formulation

- Problem: Compute robust skin/armor thickness under extreme uncertainty
- Approach: Use a version of Wald's maxmin approach, info-gap theory (Ben-Haim 2001)

  - Does not require definite *a priori* uncertainty quantification
  - Problem is expressed as nested uncertainty analyses, e.g.,

$$U(\alpha, \hat{u}) = \{u(t) : |u(t) - \hat{u}(t)| \leq \alpha\varphi(t)\}, \alpha \geq 0,$$

  - where $U(\alpha, \hat{u})$ is the set of functions $u(t)$ that deviate for the nominal function $|\hat{u}(t)|$by no more than $\alpha\varphi(t)$ for some known envelope function $\varphi(t)$; α is the uncertainty parameter

Cont.

- For a vector of design variables x, robustness is expressed as

$$\hat{\alpha}(x) = \max\{\alpha\text{:minimal requirements are always satisfied}\}$$

- Robustness = resistance to uncertainty; optimization not with respect to performance; satisficing at critical survival conditions
- If success is measured by a merit function M and m_c is a critical merit value, then robustness is

$$\hat{\alpha}(x, m_c) = \max\{\alpha\colon (\min_{u \in U(\alpha,\hat{u})} M(x, u)) \geq m_c\}$$

- For multidisciplinary (multiobjective) performance, robustness is

$$\hat{\alpha}(x, m_c) = \max\{\alpha\colon (\min_{u \in U(\alpha,\hat{u})} M_i(x, u)) \geq m_{c,i}, i = 1, \ldots, N\}$$

- where m_c is a vector of critical merits
- Example: Use the FAA penetration equation $V_{50} = \sqrt{\frac{2LCSt^2}{m\cos^2\theta}}$ to compute robust skin/armor thickness

Cont.

- Choose the thickness t, so that $x = \frac{1}{V_{50}}$ is acceptably small (or V_{50} is acceptably large), subject to uncertain mass m, i.e.

$$x \leq x_c \text{ for some } x_c > 0$$

- The nominal mass $\hat{m}$ is known (doesn't have to be; may be a distribution)
- The actual mass m is unknown
- The mass uncertainty is represented by a model

$$U(\alpha, \hat{m}) = \{m \,|\, |m - \hat{m}| \leq \alpha\}, \alpha > 0$$

- Robustness $\hat{\alpha}$ = the greatest value of α for which the performance requirement $x \leq x_c$ holds

Example, cont.

- To compute robustness:
- Evaluate maximum x, up to uncertainty α

$$\max_{m\in U(\alpha,\widehat{m})} x = \sqrt{\frac{(\widehat{m}+\alpha)cos^2\theta}{2LS_c t^2}}$$

- Equate maximum to critical value x_c and solve for α

$$\hat{\alpha}(t,x_c) = \begin{cases} \frac{2x_c^2 LS_c t^2}{cos^2\theta} - \widehat{m}, \text{ if expression} \geq 0 \\ 0\ otherwise \end{cases}$$

- Require that $\hat{\alpha}(t,x_c) \geq \hat{\alpha}_d$, some minimum design value, i.e.,

$$\frac{2x_c^2 LS_c t^2}{cos^2\theta} - \widehat{m} \geq \hat{\alpha}_d$$

Example, cont.

- And t must satisfy

$$(*) \qquad t \geq \sqrt{\frac{(\hat{\alpha}_d + \hat{m})cos^2\theta}{2x_c^2 L S_c}}$$

- Interpretation: acceptable V_{50} is guaranteed if the value of the uncertainty parameter α is smaller than the value of the robustness threshold $\hat{\alpha}_d$, i.e., if t satisfies (*), then

$$\left|\frac{1}{V_{50}}\right| \leq |x_c| \forall m \in U(\alpha, \hat{m})$$

- The choice of threshold robustness $\hat{\alpha}_d$
  - Experience
  - Dimensional: in the example – same units; $\hat{\alpha}/\hat{m} \ll 1 \Rightarrow$design vulnerable to small mass variations; $\hat{\alpha}/\hat{m} \geq 1 \Rightarrow$insensitive to large variations in mass.
  - Based on acceptable levels of risk (evaluation of consequences; qualitative)

Concluding Remarks

- Trustworthiness of an autonomous system is broadly multidisciplinary
- Major directions in developing trustworthiness:
  - Explicit modeling of decision making (cannot make assumptions, as in human decision making)
  - Comprehensive identification of uncertainties, including design, sensors, perception, and computational tractability in a multi-agent system (lacking data, lacking safe envelope notion for machine learning)
  - Address irreducible uncertainties (unknown unknowns)
  - Define “safety envelope” for visual perception (ML)
  - V&V for ML
  - Who has the final authority in multi-agent decision-making?

Acknowledgment:

Thanks to LaRC CIF/IRAD program and NASA’s Convergent Aeronautics Solutions (CAS) Project of the Transformative Aeronautic Concepts Program (TACP) and the ATTRACTOR team.